IT3105: Project 1
Implementation and Comparison of AI Algorithms to Solve The n-Queens Problem

Aksel Reiten and Magnus Moan

September 2016

Contents

1 Part 1

1.1 Board representation

We chose to represent our board in a one-dimensional array of size n, where each index represents a column and each number represents a row in the board. We chose this representation because it would (1) allow us to store and manipulate data in an efficient way with respect to time and space constraints and (2) easily guarantee absence of conflicts in both rows and columns.

1.2 Heuristic function

We are doing two different cut-offs in the search space to make the backtracking search go faster. The first thing we do is that we make sure that there never will be placed a queen in a row if there's already a queen in that row. This is

done by maintaining a set with currently unused rows.
We also ensure that we never place a queen in a position which leads to a direct diagonal conflict. This is done by checking the "slope" between the position we are testing and all the queens already placed. If we look at the chess board as a grid we observe that if any two queens have a line with slope 1 between them, they are on the same diagonal. Two queens, named A and B, are on the same diagonal if, and only if, one of the two following equalities hold:

$$row_A + col_A = row_B + col_B \tag{1}$$

$$col_A - row_A = col_B - row_B \tag{2}$$

Notice that these equations can be switched around in multiple ways. We chose this exact way since we think this is the least computational expensive way of performing this check.

2 Part 2

2.1 Solutions

With 6 8 10 12 as starting positions we got the following results on a Intel Xeon 2.67GHz machine:

N	# of solutions	Time used (seconds)
14	77	0.01
16	1917	0.25
18	39804	7.98
20	1273580	367.24

Table 1: Results backtracking

3 Part 3

3.1 Solutions

3.2 Tabu Search

3.3 Simulated Annealing

3.4 Genetic Algorithm

3.5 Comparison

	# of solutions			Time (seconds)		
N	TS	SA	GA	TS	SA	GA
18	11120	3190		150.00	150.01	
20	8872	2312		150.00	150.01	
24	6125	1255		150.01	150.04	
30	3235	594		150.03	150.30	

Table 2: Results Tabu Search, Simulated Annealing, Genetic Algorithm